NAG Library Routine Document

G05PZF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G05PZF generates a random two-way table.

2 Specification

```
SUBROUTINE G05PZF (MODE, NROW, NCOL, TOTR, TOTC, R, LR, STATE, X, LDX,     &
                   IFAIL)
INTEGER            MODE, NROW, NCOL, TOTR(NROW), TOTC(NCOL), LR,           &
                   STATE(*), X(LDX,NCOL), LDX, IFAIL
REAL (KIND=nag_wp) R(LR)
```

3 Description

Given m row totals R_i and n column totals C_j (with $\sum_{i=1}^{m} R_i = \sum_{j=1}^{n} C_j = T$, say), G05PZF will generate a pseudorandom two-way table of integers such that the row and column totals are satisfied.

The method used is based on that described by Patefield (1981) which is most efficient when T is large relative to the number of table entries $m \times n$ (i.e., $T > 2mn$). Entries are generated one row at a time and one entry at a time within a row. Each entry is generated using the conditional probability distribution for that entry given the entries in the previous rows and the previous entries in the same row.

A reference vector is used to store computed values that can be reused in the generation of new tables with the same row and column totals. G05PZF can be called to simply set up the reference vector, or to generate a two-way table using a reference vector set up in a previous call, or it can combine both functions in a single call.

One of the initialization routines G05KFF (for a repeatable sequence if computed sequentially) or G05KGF (for a non-repeatable sequence) must be called prior to the first call to G05PZF.

4 References

Patefield W M (1981) An efficient method of generating $R \times C$ tables with given row and column totals *Appl. Stats.* **30** 91–97

5 Arguments

1: MODE – INTEGER *Input*

On entry: a code for selecting the operation to be performed by the routine.

MODE $= 0$
: Set up reference vector only.

MODE $= 1$
: Generate two-way table using reference vector set up in a prior call to G05PZF.

MODE $= 2$
: Set up reference vector and generate two-way table.

Constraint: MODE $= 0$, 1 or 2.

2: NROW – INTEGER *Input*

On entry: m, the number of rows in the table.

Constraint: NROW ≥ 2.

3: NCOL – INTEGER *Input*

On entry: n, the number of columns in the table.

Constraint: NCOL ≥ 2.

4: TOTR(NROW) – INTEGER array *Input*

On entry: the m row totals, R_i, for $i = 1, 2, \ldots, m$.

Constraints:

$$\text{TOTR}(i) \geq 0, \text{ for } i = 1, 2, \ldots, m;$$
$$\sum_{i=1}^{m} \text{TOTR}(i) = \sum_{j=1}^{n} \text{TOTC}(j);$$
$$\textstyle\sum_i \text{TOTR}(i) > 0, \text{ for } i = 1, 2, \ldots, m.$$

5: TOTC(NCOL) – INTEGER array *Input*

On entry: the n column totals, C_j, for $j = 1, 2, \ldots, n$.

Constraints:

$$\text{TOTC}(j) \geq 0, \text{ for } j = 1, 2, \ldots, n;$$
$$\sum_{j=1}^{n} \text{TOTC}(j) = \sum_{i=1}^{m} \text{TOTR}(i).$$

6: R(LR) – REAL (KIND=nag_wp) array *Communication Array*

On entry: if MODE $= 1$, the reference vector from the previous call to G05PZF.

On exit: the reference vector.

7: LR – INTEGER *Input*

On entry: the dimension of the array R as declared in the (sub)program from which G05PZF is called.

Constraint: $\text{LR} \geq \displaystyle\sum_{i=1}^{m} \text{TOTR}(i) + 5$.

8: STATE($*$) – INTEGER array *Communication Array*

Note: the actual argument supplied **must** be the array STATE supplied to the initialization routines G05KFF or G05KGF.

On entry: contains information on the selected base generator and its current state.

On exit: contains updated information on the state of the generator.

9: X(LDX, NCOL) – INTEGER array *Output*

On exit: if MODE $= 1$ or 2, a pseudorandom two-way m by n table, X, with element $\text{X}(i, j)$ containing the (i, j)th entry in the table such that $\displaystyle\sum_{i=1}^{m} \text{X}(i, j) = \text{TOTC}(j)$ and $\displaystyle\sum_{j=1}^{n} \text{X}(i, j) = \text{TOTR}(i)$

10: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G05PZF is called.

Constraint: LDX $\geq$ NROW.

11: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, MODE $= \langle value \rangle$.
Constraint: MODE $= 0$, 1 or 2.

IFAIL $= 2$

On entry, NROW $= \langle value \rangle$.
Constraint: NROW ≥ 2.

IFAIL $= 3$

On entry, NCOL $= \langle value \rangle$.
Constraint: NCOL ≥ 2.

IFAIL $= 4$

On entry, at least one element of TOTR is negative or TOTR sums to zero.

IFAIL $= 5$

On entry, TOTC has at least one negative element.

IFAIL $= 6$

NROW or NCOL is not the same as when R was set up in a previous call.
Previous value of NROW $= \langle value \rangle$ and NROW $= \langle value \rangle$.
Previous value of NCOL $= \langle value \rangle$ and NCOL $= \langle value \rangle$.

IFAIL $= 7$

On entry, LR is not large enough, LR $= \langle value \rangle$: minimum length required $= \langle value \rangle$.

IFAIL $= 8$

On entry, STATE vector has been corrupted or not initialized.

IFAIL $= 10$

On entry, LDX $= \langle value \rangle$ and NROW $= \langle value \rangle$.
Constraint: LDX $\geq$ NROW.

IFAIL $= 15$

On entry, the arrays TOTR and TOTC do not sum to the same total: TOTR array total is $\langle value \rangle$, TOTC array total is $\langle value \rangle$.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

None.

8 Parallelism and Performance

G05PZF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

None.

10 Example

Following initialization of the pseudorandom number generator by a call to G05KFF, this example generates and prints a 4 by 3 two-way table, with row totals of 9, 11, 7 and 23 respectively, and column totals of 16, 17 and 17 respectively.

10.1 Program Text

```
    Program g05pzfe

!     G05PZF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g05kff, g05pzf, nag_wp, x04eaf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
```

```
      Integer, Parameter               :: lseed = 1, nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: genid, ifail, ldx, lr, lstate, mode, &
                                          ncol, nrow, subid
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: r(:)
      Integer                          :: seed(lseed)
      Integer, Allocatable             :: state(:), totc(:), totr(:), x(:,:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: sum
!     .. Executable Statements ..
      Write (nout,*) 'G05PZF Example Program Results'
      Write (nout,*)
      Flush (nout)

!     Skip heading in data file
      Read (nin,*)

!     Read in the base generator information and seed
      Read (nin,*) genid, subid, seed(1)

!     Initial call to initializer to get size of STATE array
      lstate = 0
      Allocate (state(lstate))
      ifail = 0
      Call g05kff(genid,subid,seed,lseed,state,lstate,ifail)

!     Reallocate STATE
      Deallocate (state)
      Allocate (state(lstate))

!     Initialize the generator to a repeatable sequence
      ifail = 0
      Call g05kff(genid,subid,seed,lseed,state,lstate,ifail)

!     Read in the problem size
      Read (nin,*) nrow, ncol

      ldx = nrow
      Allocate (totr(nrow),totc(ncol),x(ldx,ncol))

!     Read in row and column totals
      Read (nin,*) totr(1:nrow)
      Read (nin,*) totc(1:ncol)

      lr = sum(totr(1:nrow)) + 5
      Allocate (r(lr))

!     Using a single call to G05PZF, so set up reference vector
!     and generate values in one go
      mode = 2

!     Generate the random table
      ifail = 0
      Call g05pzf(mode,nrow,ncol,totr,totc,r,lr,state,x,ldx,ifail)

!     Display the results
      ifail = 0
      Call x04eaf('General',' ',nrow,ncol,x,ldx,'Random Table',ifail)
      Write (nout,*)
      Write (nout,Fmt=99999) 'Supplied row totals:', totr(1:nrow)
      Write (nout,Fmt=99999) 'Supplied column totals:', totc(1:ncol)
99999 Format (1X,A,/,(2X,4I10))
    End Program g05pzfe
```

10.2 Program Data

```
G05PZF Example Program Data
1  1  1762543      :: GENID,SUBID,SEED(1)
4  3               :: NROW,NCOL
9   11   7   23    :: TOTR
16  17  17         :: TOTC
```

10.3 Program Results

```
 G05PZF Example Program Results

 Random Table
    1 2 3
 1  2 4 3
 2  6 1 4
 3  2 4 1
 4  6 8 9

 Supplied row totals:
           9        11         7        23
 Supplied column totals:
          16        17        17
```